SERVICE SYSTEM IN A NETWORK

BACKGROUND OF THE INVENTION

Field of the Invention:

[0001] The present invention generally pertains to IP-based communication networks, and in particular to supplementary services for next generation IP-based communication networks.

Discussion of the Related Art:

[0002] In IP-based communication networks (for telecommunication and/or data), Network equipment vendors will add value to their products by providing distributed Network Services (e.g. QoS, security, reliability, roaming, remote access, . . .) to enhance basic-IP connectivity.

[0003] Independent third party companies will provide Network Applications (e.g., communication, collaborative working, e-commerce, on-line gaming, distance learning, etc.) for clients/servers that transparently use the network infrastructure as a mere transport facility based on open standards and protocols.

[0004] The aforementioned facts require a flexible architecture for the provision and execution of user services in IP-based networks for telecommunication and data (converged networks).

[0005] In existing solutions for user service provisioning, e.g. in telecommunication networks like the GSTN (Global Switched Telephony Network), the intelligence for user service provision and execution are unified and centralized in a few network entities, such as in the Local Exchange LX approach (providing "switch-based services") the service processing is tightly coupled with normal call processing or in the Intelligent Network (IN) approach the call processing is separated. The combination of the Service Switching Function (SSF) and Service Control

Function (SCF) work to process services by detection of traversal of points in a call and by manipulation of the call state.

[0006] In the GSTN, said user services are called supplementary services. In general, these supplementary services automate procedures that people carry out when using "basic" telephony. In other words, supplementary services are features that make basic telephony calls more convenient.

[0007] Next generation IP-based networks will have two important characteristics, namely, mobility of users (with or without their terminals), and a wealth of intelligent terminals (PC, PDA, laptop, mobile phone with Java Virtual Machine (JVM), etc.) These characteristics drive requirements for the benefit of the user and the network.

[0008] Roaming of nomadic users combined with transparent co-migration of the user's Virtual Home Environment (VHE) will be desired. The VHE is a user's personal set of subscribed applications, persona options of how these applications are configured and bundled into a user services package, and the user's personal choice of payment for the use of these services.

[0009] In addition, compatibility with intelligent communication terminals like a PC, PDA, lap/palmtop, mobile phone with Java Virtual Machine (JVM) etc. will also be desired.

[0010] A user may not be limited to a single type of intelligent terminal but instead will want to use what is most appropriate or what is available at a particular location. Despite changing terminals at will, the user always wants to have the same services available regardless of the particular end device used.

[0011] Due to the flexible nature of IP-based networks, the number of possible applications and user services running on intelligent terminals is enormous. Programmers are constantly implementing new ideas based on inexpensive, off-the-shelf platforms for intelligent terminals (e.g. PCs).

[0012] An architecture for the provision of user services must be able to cope with ever changing applications and user services for even more rapidly evolving terminal hardware. As the network edges are changing so fast, it will become impossible to provide and execute user services in the core of the network. Core equipment is subject to more stringent requirements than consumer-grade, low-cost edge equipment (terminals) and hence more expensive. Creation of applications and corresponding user services (supplementary services in the case of Internet Telephony) on core equipment would technologically and economically not be able to keep up with the rapid creation of new applications and user services on inexpensive edge equipment.

[0013] The principle idea of the invention is the function split in realizing services: providing of user services (e.g. enabling, profiling, distribution, administering and billing of user services) remains in the core of the network, while the execution of user services is delegated to intelligent terminals at the network edges.

[0014] Since by now terminals have become powerful processing units anyway, it is economically and technically reasonable to delegate the execution of user services to intelligent terminals at the network edges. However providing of user services (e.g. enabling, profiling, distribution, administering and billing of user services) remains in the core of the network.

[0015] Technically this is achieved by a corresponding client/server architecture with at least one central application repository server keeping copies of application programs that can be downloaded on-demand into the user's terminals.

[0016] This architecture for the provision of user services in next generation IP-based networks for telecommunication and/or data meets all the aforementioned user and network requirements.

[0017] The invention optimally makes use of the two sources of intelligence that will be present in such networks, namely, distributed intelligence in network elements (such as routers and switches) adding value to basic IP-connectivity, and intelligence in end systems (clients, servers) allowing for a wealth of advanced applications.

[0018] This is a new notion of "network intelligence" different from the interpretation in classical IN for the GSTN where intelligence solely rests in central entities. There the terminals are just dumb telephony sets without any processing power.

[0019] With terminals becoming powerful processing units due to technical evolution, an embodiment of the invention includes a corresponding application execution environment (application execution component) that works across a wide range of intelligent terminals and enables the technical feasibility of delegating service execution to a spectrum of different intelligent terminals.

[0020] User services are services to the end user resulting from the interworking of network-aware applications which operate above IP-Model Layer 3 (typically even above IP-Model Layer 5) in intelligent terminals at the edges of the network, e.g. e-mail, WWW, Buddy Lists, Internet Telephony, etc.

[0021] Network services add value to the basic service of "IP-transport." These services are provided at IP-Model Layers 2 & 3 by network elements which operate inside the network, e.g. QoS, security, VPN, etc.

[0022] The invention is concerned with an architecture for the provision of user services to IP-based intelligent terminals, not with the provision of network services.

[0023] For the purpose of the invention it is sufficient to treat the IP-based network as an entity providing IP-connectivity only. The network is considered to be a pure transport network offering no specific network services.

SUMMARY OF THE INVENTION

[0024] It is an object of the present invention to offer a system that is not concerned with the fundamental processes involved in making a call, which functionality is provided by the underlying Internet Telephony package used.

[0025] It is another object of the invention to provide a standard interface that enables supplementary services to interact with all telephony packages in the same manner, without the need for any package-specific code in the services and whilst keeping the service/package interaction as simple as possible.

[0026] It is an additional object of the invention to offer the ability to download a user-specific set of small application programs from a central server into the user's end system.

[0027] It is yet another object of the invention to limit the supplementary services a user has access to and to charge the user for the services used.

[0028] It is yet a further object of the invention to offer the ability to execute the downloaded programs that implement a set of supplementary services on the user's machine, in a secure environment and regardless of the machine platform.

[0029] It is yet an additional object of the invention to offer an overall control process that allows multiple, potentially conflicting, supplementary services to be executing at the same time without problem. The user is able to assign and change the relative priority of services within the service set and thus modify how potential conflicts are to be resolved.

[0030] It is still another object of the invention to offer the ability for individual supplementary services in the downloaded service set to be configured by the user. This configuration includes setting service-specific parameters, setting the relative priority of services within the set of supplementary services and the ability to enable and disable individual services on a temporary basis.

BRIEF DESCRIPTION OF THE DRAWINGS

[0031] Figure 1 shows elements of an architecture for the provision of use services in next generation networks;

[0032] Figure 2 shows architecture for the provision of user services in "idle state": ready to enable a user with his specific service profile when he connects with his intelligent terminal to a network access point;

[0033] Figure 3 shows step-by-step service delivery: (user authentication ("user ID-to-service profile" mapping ("service profile-to-applications bundle" mapping (delivery of user specific applications bundle;

[0034] Figure 4 shows service delivery completed: the user is ready to use his personal profile of user services;

[0035] Figure 5 shows detailed view of the Application Execution Environment (AXE);

[0036] Figure 6 shows minimal architecture for Internet Telephony; and

[0037] Figure 7 shows architecture for supplementary services in Internet Telephony.

DETAILED DESCRIPTION OF THE PRESENTLY PREFERRED EMBODIMENTS

[0038] The general architecture involves only a small number of network entities as illustrated in Fig. 1, which shows a network entity with its own IP address, memory, and some sort of CPU. For the near future, these are the numerous workstations, PCs, lap-&palmtops, IP-(video) phone sets, TV set-top boxes, mobile communication devices that run some sort of operating system and become-following Moore's Law-equipped with ever more powerful CPU's every 18 months. Each of these terminals understands some variant of Java. Today, the Java Virtual Machine

(JVM) is already a standard part of most operating systems or gets molded into dedicated silicon chips for small devices like mobile communication devices. There is and will be a wealth of terminals available to the user. Each of them far more "intelligent" than the standard phone handset of today. The proposed architecture will therefore make use of this "free and idle intelligence" by delegating execution of user services to it.

[0039] Depending on the particular intelligent terminal considered, this is a LAN connector, a dial-up POP of an ISP, a lap/palmtop docking unit, a base station for mobile communications, etc. In the proposed architecture the sole purpose of the network access point is to provide IP-network connectivity.

[0040] Given the possibility of users roaming from access point to access point, getting dynamically assigned IP-address while at the same time switching between PC, laptop, mobile communication device, etc., the access does not handle advanced tasks such as authentication or any other user specific things.

[0041] If the execution of user services is delegated to the intelligent terminals, and the network access point provides just pure IP-connectivity, then the intelligence beyond mere execution of user services must be kept elsewhere. This is the User Service Merchant. The User Service Merchant acts as a central source for user specific information like user ID, authentication data, profile of subscribed user services, billing information, etc. It also keeps "master copies" of the application executables in a dedicated repository (see next section for a detailed explanation).

[0042] The User Service Merchant provides all the logic necessary to provision services to the user but does not actually perform the services.

[0043] In the existing classical IN for the GSTN, the intelligence for service provision and execution are unified and centralized in a few network entities. This is opposite to the solution proposed here, where the intelligence is split and decentralized: provision of user services is handled centrally by the User Service

Merchant, the actual execution of user services is delegated to the distributed intelligent terminals.

[0044] Fig. 2 shows the architecture in "idle state". The User Service Merchant is connected to the network, the user's intelligent terminal (PC, laptop, mobile communicate device, . . .) may or may not be connected to the network via a network access point. At this point, the User Service Merchant considers the terminal as "off-line" (in Fig. 2 represented in italics) in either case.

[0045] The User Service Merchant keeps a list of user IDs (UID). An UID uniquely identifies a subscriber and indirectly associates him with a certain bundle of pre-selected applications. The information about the exact selection of applications is kept as an UID-associated Service Profile for each individual subscriber on the User Service Merchant.

[0046] The User Service Merchant also maintains an Application Repository which basically is a collection of application programs stored in a suitable database system. Each program either implements a self-contained user service, or some value-adding supplementary feature for such a service (example: the self-contained service could be "text-only Web browsing" for mobile communication appliances, the supplementary feature could be "optional graphics support for Web browsing" with the same appliance). The programs are implemented such that an instance of them can be downloaded on demand from the repository into the user's intelligent terminal for execution. This may be achieved by technologies such as Java.

[0047] A user can (un)subscribe to any of the available application programs in the repository by correspondingly changing the configuration of his UID-associated Service Profile. This update of the Service Profile is propagated to the User Service Merchant for permanent storage when the user terminates his session or when he takes his intelligent terminal off-line. The Service Profile is not stored on the user's intelligent terminal to allow for easy roaming or switching between different intelligent terminals. With the Service Profile kept centrally on the User Service

Merchant the user always gets the same Service Profile regardless whether he accesses the network e.g. from his office PC, from a public information kiosk, from a mobile communication device, or from home. Changes to the Service Profile made from one location/terminal will be available automatically at any other location/terminal.

[0048] This concept of a Service Profile transparently migrating with the user across different locations and end systems is referred to as Virtual Home Environment (VHE). Service delivery with this type of architecture is shown in Fig. 3. Steps involved with this type of architecture include: User Authentication, UID-to-Service Profile Mapping, Service Profile-to-Applications Bundle Mapping, and Delivery of User Specific Applications Bundle.

User Authentication:

[0049] Before a user can use any of the services he has to authenticate himself by providing his UID ① (and related password/PIN) to the Service Merchant. In the case of a permanently connected PC this can be done by clicking on a corresponding button. In the case of a laptop or a mobile communication device, this can be done automatically when it gets connected to the Network Access Point.

UID-to-Service Provide Mapping:

[0050] After the UID (dark gray triangle) has been verified ②, it is mapped to the corresponding Service Profile of the user. The Service Profile (light gray blobs) represents the user's personalized applications bundle compiled from the total set of available applications (dark gray blobs) in the Application Repository. Via the Service Profile the user can create his very personal VHE. The provide is not static but can be customized and administered by the user directly from his intelligent terminal.

Service Profile-to Applications Bundle Mapping:

[0051] Based on the information in the user's Service Profile, the User Service Merchant identifies a subset of the applications available (3) in the Application Repository. The identified subset (circled dark gray blobs) is the personalized bundle of user services.

Delivery of User Specific Applications Bundle:

[0052] The final step (4) is to download the selected bundle of applications over the network into the user's intelligent terminal. The Java software technology was designed with exactly this download capability of programs in mind. Related issues such as platform independence, efficiency and security when transmitting code over a network area are an integral part of the Java design. Together with its increasing availability in intelligent terminals, this makes Java the prime candidate to support the type of architecture for the delivery of user services discussed in this document.

[0053] Fig. 4 shows the architecture in the "ready state", the final stage of the service provisioning process. The programs enabling the subscribed user services have been downloaded into his intelligent terminal and await their invocation and execution there.

[0054] As end systems become equipped with ever more powerful processors, it is just reasonable to exploit this power by having the services execute on the end systems rather than some central network entity. Restraining themselves to service delivery issues rather than execution central network entities face less stringent scalability problems.

[0055] Furthermore, with this approach central network entities like the User Service Merchant are out of the loop of rapid technology changes in intelligent terminals like PCs, laptops, mobile communication devices, etc. If service specific details regarding the intelligent terminals change, only the affected modules of the

programs in the Application Repository have to be adapted. The User Service Merchant as a whole with its user administration, authentication, application repository & delivery functions is not affected at all. This modular approach to the provision of user services is nicely supported by the dynamic loading of required classes available in Java. Not even the whole program has to be replaced but just the affected class file.

[0056] With execution of user services delegated to the intelligent terminals in the form of downloadable application programs, it becomes important to devise a corresponding Application execution environment that works across a wide range of technologically different intelligent terminals. The Application execution environment is therefore implemented as a virtual machine, for example as a java virtual machine.

[0057] The virtual machine has to be ported to each supported terminal only once. To the applications the virtual machine looks the same on all supported terminals.

[0058] Fig. 5 shows a correspondingly detailed view of an intelligent terminal for the case of a PC running the Windows operating system by Microsoft.

[0059] The downloaded application programs that implement a particular user service are Java applets (dark gray blobs) which are executed in the Application Execution Environment (AXE) on the intelligent terminal. Communication to and from the user is done via a graphical user interface (GUI) shared by all user services. The downloaded programs are confined within the AXE (so-called 'sandbox' approach) and do not communicate to lower software layers other than through a specific AXE API. The platform dependent connection to the Windows DLLs is achieved by a Java/COM interface taking AXE API commands as input. This interface has to be written only once (e.g. in C++) for each supported intelligent terminal. The interface forms the specific link between the platform independent

under services implemented in Java and the actual operating system and hardware of the terminal.

[0060] This evolves the general architecture further in the case of supplementary services for Internet Telephony.

[0061] Today, Internet Telephony is still lacking the elaborate set of supplementary services offered by the Intelligent Network (IN) approach in the Global Switched Telephone Network (GSTN). Internet Telephony is mainly used to provide an analogy for telephony over the GSTN.

[0062] Fig. 6 shows a minimal configuration by which Internet Telephony (and, in particular, Microsoft (MS) NetMeeting) calls are arranged.

[0063] In this arrangement, there are three PCs needed, namely, the Internet Locator Service (ILS) Server, and the originating and terminating end stations. The end stations are involved actively in the communication session, transcoding audio streams (and, optionally, video streams or data) into a form that can be carried over the Internet. The ILS (Internet Locator Service) Server acts as a directory server in the configuration. This node stores information on the users who are "logged in". Via the ILS users are associated with an Internet host, and that Internet host is running an Internet Telephony application and so is ready to respond to incoming call requests. When a user asks for an Internet Telephony call to be originated, an address query is sent to the directory server with the "alias" (usually the email identifier) of their intended correspondent. The ILS then responds with the user's currently associated IP-address.

[0064] With Internet Telephony, a number of distinctive features for supplementary services become possible that are not available in the case of GSTN.

[0065] The important point to note here is that user services, in this particular case supplementary services for Internet Telephony, are associated with applications running on intelligent terminals, not some sort of central network entity.

[0066] Use of the Internet allows a great deal of complexity in the provision of supplementary services to be avoided. Many of the services that require user interaction in the GSTN require temporary connections to be made to specialized resources (such as announcement units and DTMF tone decoders), and for any captured information to be sent onwards to service processors using separate data links. All of this is avoided because, on the Internet, one can assume that a user is associated with an end station that allows involved text and graphics to be displayed, and that the end station is capable of transferring complex messages directly, rather than being restricted to a simple keypad.

[0067] There is another implication of the use of the Internet when providing supplementary services. Internet hosts are computers, and so will almost certainly have powerful processors. This opens the possibility of carrying out much of the processing on the end stations rather than relying on the network nodes (such as the LX or the IN) as is required in the GSTN due to the lack of local intelligence in the POTS telephones.

[0068] Combination of basic Internet Telephony with other Internet applications such as email enables yet another completely different set of truly advanced supplementary services.

[0069] The architecture proposed here implements and extends the more general architecture for the case of supplementary services for Internet Telephony.

[0070] The architecture is concerned with an architecture to support the provision of supplementary services for Internet Telephony. It is not concerned with providing a solution for basic Internet Telephony calls. This functionality will be taken from some off-the-shelf Internet Telephony application package like Microsoft (MS) NetMeeting.

[0071] The proposed architecture contains an infrastructure for distributing service logic (in the form of Java applets) to the end stations, and for allowing the service logic and it's management infrastructure to operate in a distributed fashion (both in the end stations and in separate servers). The distributed service logic supports and interworks with the aforementioned Internet Telephony application package.

[0072] Fig. 7 shows the general arrangement of systems providing Internet Telephony, with one of the end stations (shown at the right) also having the enhancements to support supplementary services for Internet Telephony. The architecture contains the two end stations discussed above, each running off-the-shelf MS NetMeeting Internet Telephony application packages, where at least one application is enhanced with the J/Direct Interface and the Application Execution Environment (AXE). The server offers directory services for translation of a symbolic user alias (e.g. email address) to the user's current IP address. The ILS is used to allow for easy to remember Telephony "numbers" and hence supports convenient call setup by the user.

[0073] The User Service Merchant acts as a repository for the supplementary services for Internet Telephony. The services are stored as pre-compiled programs (in the form of Java applets), each implementing a specific service logic. On demand by the user, these service logic units are downloaded into the user's end stations for execution there. The User Service Merchant also keeps track of which user is entitled to use which supplementary service.

[0074] When a user requests a particular Internet Telephony supplementary service, the Java applet with the corresponding service logic is downloaded from the Service Merchant into the AXE which sits on top of the user's Internet Telephony application package. The applet runs in the AXE and drives the telephony application package according to the applet's service logic through the J/Direct interface. The J/Direct interface allows for the service logic to access the Internet Telephony libraries of the underlying application package.

[0075] Given the architecture of the present invention, a wealth of supplementary services for Internet Telephony can be provided. The set of sample Internet Telephony supplementary services listed below are aimed at combining email and Internet Telephony. The services can be grouped into "originating" and "terminating" services depending on whether the caller or callee uses them.

[0076] This originating service is triggered when the callee is already engaged in a call. Although the user may have rejected the offered call explicitly, the result (inability to communicate) is the same, and this service suffices in both cases. The caller is asked whether or not he wants to send an email instead. If the caller answers in the affirmative, then the rest of the service proceeds. The conference and call objects created by NetMeeting hold information on the called user alias, and this is used to make a template for an email to be filled in by the caller. This template is displayed on the user interface, and, on entry of the email content, is submitted for transmission. By implication, the service logic must include a small email client to send the email to an SMTP server.

[0077] As in the previous case, this supplementary service is triggered when a call rejection has been received from the remote callee. In this case, however, it will display a user interface dialogue asking the calling subscriber if they would like the system to retry the callee's address at regular intervals. If the answer returned is "yes", then the service proceeds. A timer is started, and on its expiry, the call is re-initiated. After that, the service monitors the call state, and, if the call is successful, the user is notified. If it again fails, the timer is reset, and the sequence repeats. A choice on the user interface to be presented to allow for canceling of the sequence once it has could be an additional feature.

[0078] This terminating supplementary service is used to process calls delivered to the end station in terms of the caller's identity. In this case, it will be used to "auto-reject" call offers from people not on a list of names generated as a preservice procedure. The list can contain explicit caller IDs or can include mechanisms to reject calls from people who have arranged their Internet Telephony

client not to pass on their identity (by the expedient of not registering with an ILS server prior to making the call, in the case of NetMeeting).

[0079] This terminating supplementary service allows a subscriber (callee) to be informed by email of the identifies of the people who have called him whilst he was unable (or unwilling) to answer his Internet Telephony program. The supplementary service will extract the name of the caller from the call data that NetMeeting provides (as well as making a note of the current time), and then construct an email describing this information. Once this is ready, it will connect to the callee's SMTP server, will send the text it has created, and then finally terminate the connection. Enhancements of this service could be used to send other kinds of notification (such as, for example, contacting a Web-based gateway to the GSM Short Message Service (SMS) and sending an SMS message with the caller's identity).

[0080] Although modifications and changes may be suggested by those skilled in the art to which this invention pertains, it is the intention of the inventors to embody within the patent warranted hereon all changes and modifications that may reasonably and properly come under the scope of their contribution to the art